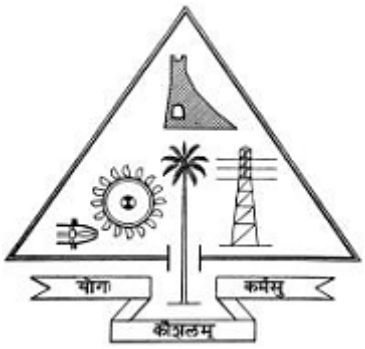
A REAL-TIME SYSTEM FOR TENNIS BALL TRACKING AND TRAJECTORY PREDICTION

Thesis submitted in partial fulfillment of the requirements for the award of the degree of ***Master of Computer Applications*** *of the* ***APJ Abdul Kalam Technological University***

submitted by

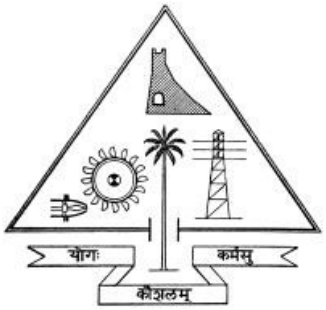
SAFEER SHAHUL
(TCR19MCA021)



DEPARTMENT OF COMPUTER APPLICATIONS
GOVERNMENT ENGINEERING COLLEGE
THRISSUR - 680009

MAY 2022

DEPARTMENT OF COMPUTER APPLICATIONS
GOVERNMENT ENGINEERING COLLEGE, THRISSUR
THRISSUR, KERALA STATE, PIN 680009



CERTIFICATE

This is to certify that the main project titled **"A REAL-TIME SYSTEM FOR TENNIS BALL TRACKING AND TRAJECTORY PREDICTION"** *is a bonafide work done by* **SAFEER SHAHUL (TCR19MCA021)** *under my supervision and guidance, and is submitted in May 2022 in partial fulfillment of the requirements for the award of the Degree of Master of Computer Applications from APJ Abdul Kalam Technological University(KTU).*

Shehin Shams P	Soumia Chandran M	Dr. Sminesh C N
Project Guide	**Project Coordinator**	**Head of Department**

Place : THRISSUR
Date : 13-05-2022

DECLARATION

I hereby declare that the main project named, **A Real-Time System for Tennis Ball Tracking and Trajectory Prediction**, is my own work and that, to the best of my knowledge and belief, it contains no material previously published another person nor material which has been accepted for the award of any other degree or course of the university or any other institute of higher learning, except where due acknowledgement and reference has been made in the text.

Place : THRISSUR Signature

Date : 13-05-2022 **SAFEER SHAHUL(TCR19MCA021)**

ACKNOWLEDGEMENT

I would like to thank Computer Application Department of GEC Thrissur, for giving me this opportunity to pursue this project and successfully complete it.

I am highly indebted to my guide **Prof. Shehin Shams P** and project coordinator **Prof. Soumia Chandran M** for their guidance and constant supervision as well as for providing necessary information regarding the project and also for her support in completing the project.

I also express special thanks to the Head of the Computer Applications Department, **Dr. Sminesh C N** for his keen support and consistent encouragement in our academic activities

I sincerely thank all other faculties of MCA department for guiding through the processes involved in the project.

ABSTRACT

Sports are significantly more than just games for millions of individuals throughout the world. They're an important element of their identity and inspire passions both on and off the field. Sports, on the other hand, are a business for teams and players, attracting investments, profit, and expensive contracts that turn previously unknown athletes into millionaires. And it's a booming business. With so much on the line, it's no surprise that leagues and teams are doing everything they can to keep supporters happy while also gaining a competitive advantage, with artificial intelligence (AI) and robotics technology driving the transformation.

The project is aimed to detect the tennis ball from live video, Track the tennis ball and predict the tennis ball. As the use of sports robotics grows, new concerns emerge. When we need to implement a tennis-playing robot, for example. The issues listed above must be resolved. Furthermore, because to its drawbacks, we cannot always employ a two-dimensional camera to solve this type of problem. This problem is solved with a two-dimensional camera. This idea could be implemented in the future with a three-dimensional camera or by converting two-dimensional coordinates to three-dimensional coordinates.

CONTENTS

LIST OF FIGURES

NOMENCLATURE

FPS	Frames per second
mAP	Mean Average Precision
RCNN	Region-Based Convolutional Neural Network
SSD	Single Shot Detector (Object Detection)

CHAPTER 1
INTRODUCTION

In recent years, sports robots have been a hot study topic. The most significant technology for tennis robots are ball tracking and trajectory prediction. Previous research efforts resulted in the development of several methodologies, which can be split into two categories: physical models and machine learning. The former makes use of algorithms that take gravity, air resistance, the Magnus effect, and elastic impact into account. Estimating these external forces, on the other hand, necessitates high sample frequencies, which can only be accomplished with high-efficiency imaging equipment. To learn the flight trajectories of tennis balls, this study used machine learning.This project deals with the real-time prediction of a tennis ball trajectory in two dimensions.

Here the object detection is done using YOLOv5 custom object detection model which divides the images into a grid system. The Yolo model allows faster detection real-time custom object detection compares to other machine learningh models; like, RCNN and SSD. And when comparing RCNN, SSD and Yolo models, SSD possess more accuracy and Yolo model (From YoloV3) possess faster detection between them. The data set for YOLOv5 contains 1220 images and the training valid split ratio is taken as 9:1.

After object detection, then comes object tracking. From the bounding box obtained from object detection which gives the x, y coordinates, the centroid is calculated. From this centroid value the x, y coordinates for different movements are analyzed and stored in a list which gives the centroid history.

With the use of OpenCV and the above-mentioned list the object trajectory is defined. Using Kalman filter, we can predict the trajectory of the object. We can also find out the speed of the object and distance to the object with this method.

As a result, the Kalman filter is the best technique for combining real-time data and estimating the state of system parameters throughout time epochs in the estimation process.

The following are the most crucial concepts to remember while utilising the Kalman filter:

- Kalman filters are discrete, meaning they rely on measurement samples taken at regular intervals.
- Kalman filters are recursive: they forecast the future based on the current state (position, velocity, acceleration, etc.). It also makes an educated prediction regarding external elements that could influence the scenario.
- Kalman filters forecast the future by taking measurements (such as via sensors) and then adjusting the state estimate based on the prediction and measurements.

This system is created to capture real-time 2D tennis ball video. It can be used in recreational tennis games. The goal of the project was to do 3D real-time detection on tennis ball videos for the Tennis Playin robot at first. However, because to a shortage of 3D cameras at our college AI lab, we decided to construct the system for 2D real-time videos instead. For this research, we employed the AI lab's ZED 2 3D camera. However, the ZED 2 is unable to detect quickly moving objects (such as a tennis ball), and the video quality produced by the ZED camera is poor. In the future, we can employ a low-cost 3D camera to complete this project for real-time 3D tennis ball videos.

CHAPTER 2
LITERATURE REVIEW

Sports Robotics and Analysis is a hot topic in these days in research. There have been various methods discussed in various papers in accordance with this topic. Some of the methods are being discussed in this chapter. A comparative study of the existing systems and the proposed system is done which proves that there are no existing 2D real-time detection, tracking and prediction system available in this topic, which makes this work unique.

A recent paper[1] has been published based on Automated ball tracking in tennis videos - Using a mixed computer vision and machine learning approach, offers a standalone algorithm for video stabilisation and tennis ball tracking. The system uses video stabilisation algorithms to stabilise unstable video before identifying ball candidates using a random forest segmentation methodology. Random forest segmentation employs robust features such as yellow colour plane intensity and phase mathematical Fourier transform. The results outperform current state-of-the-art approaches. Furthermore, their approach detects the ball even when it is partially obscured by the tennis court net or the racket. During random forest segmentation, other features such as the shape of the ball can be introduced to the feature pool.

Another paper[2] discusses a trajectory-based ball recognition and tracking framework that may be used to retrieve ball locations in basketball long shot video sequences. Using 2-D distribution analysis of the ball candidates in the x- and y-direction individually, the ball's motion characteristic is used to identify the ball's trajectory. To discern the moving items in the fore-

ground, a strong moving object segmentation method based on background removal and frame differencing is utilised, which delivers accurate results for dynamic backdrop scenarios with substantial background clutter and can survive the impacts of camera motion. The feature-based pruning of ball candidates ensures that fewer candidates be handled during trajectory processing, lowering the overall system's computing complexity. high-speed cameras, for example. According to the authors, this is the first program to offer a thorough analysis of a basketball long shot sequence in real time.

A paper for Computer Vision and Machine Learning for In-Play Tennis Analysis[3] discusses a three-layered LSTM network capable of classifying fine-grained tennis moves was described. It revealed a lot of intriguing facts. First, our network obtained good results by extracting features using the Inception neural network, which was trained on an independent dataset and did not require fine-tuning. This shows that Inception's hidden layers are a solid data representation that can be applied to a variety of applications and domains. Second, the network's categorization errors could be deduced, implying that it was learning semantically relevant data. The network performed better when trained with solely amateur or professional players rather than a mixed population, proving this theory.

Third, when trained on a mixed population, the network performed better for professional players than amateurs. One probable explanation is that professionals utilise a more refined method that results in more distinct strokes. In the future, we'd like to see if this can be used to evaluate a player's skill level. Fourth, the same network architecture correctly classified the three primary stroke types with an accuracy of 88.16 percent, outperforming an indirect inference based on finer-grained actions. This adds to the robustness and portability of the Inception features. Finally, we evaluated the suggested technique using the HMDB dataset to show how it may be applied to broad action detection tasks. We hope that this work will inspire further research into the application of deep neural networks in

the sports domain, as well as the use and creation of benchmark datasets in sports action detection.

The presentation of the first deep Neural Network for fine-grained action detection in tennis, as well as one of the first for fine-grained action identification in general, is their key contribution. Backhands, forehands, and serves are classified with 88.16 percent accuracy, while 12 finer-grained movements are classified with 47.22 percent accuracy. We also show that the network is learning semantically relevant information because most errors are interpretable and its performance varies depending on whether the player is an expert or an amateur.

Another paper[4] discusses on Ball position in BSV is a difficult problem for a variety of reasons, as evidenced by our own experimental results. They presented a trajectory-based solution for offline ball detection and tracking in BSV to overcome this challenge. The results of the experiments reveal that their algorithm is quite effective at detecting balls—it detects not only the ball, but also obstructed balls and balls that have been fused with other objects in the frame.

This method is based on two main ideas. The first is to utilise a set of ball candidates for each frame (instead of direct ball selection). This lowers the rate of misses and allows occluded and merged balls to be considered as ball contenders. The utilisation of ball trajectories to aid in ball detection is the second fundamental concept. In other words, it is more trustworthy to determine "if a candidate trajectory is a ball trajectory" rather than "whether a ball-candidate is the ball." Other technical innovations mentioned in this study that may be relevant elsewhere include the use of an anti-model technique for ball-candidate generation, which is a fault-tolerant approach.

Another work[5] proposes a small ball tracking with trajectory prediction using Deep SORT, which combines motion and frame storage with direction

information. As a result, the suggested technique performs better on the BALL dataset when tracking little balls. As a result, this work is extremely important for modern intelligent physical education.

Another research[6] suggested TrackNetV2, an improved version of TrackNet that combines VGG16 and upsampling layers with a deep learning network. TrackNetV2 inherits the advantages of TrackNet, such as network design and successive picture input, but also adds some new features. After a deep convolution network, skip connections and maintain the boundaries feature. TrackNetV2 also changed the softmax layer to a sigmoid layer to save memory by decreasing the output range and computing the model's loss with weighted cross entropy. The network's final output is a heatmap of object position. TrackNetV2 has become significantly more resource efficient and easy to train as a result of the above advancement.

Future work will entail converting the 3-in/3-out structure to a 5-in/5-out one and considering a more current loss function. Hopefully, they will be able to reach beyond real-time applications and overcome numerous backdrops problems in order to more precisely find high-speed and small objects.

CHAPTER 3
ENVIRONMENTAL STUDY

3.1 System Configuration

System configuration describe the hardware and software requirement of the system for development

3.1.1 Hardware Requirements

- Memory : 8 GB of RAM
- CPU : Intel Core i5 11th Gen or above
- Mobile device capable of recording 30 FPS or above video.
- GPU : RTX 3050 or above

3.1.2 Software Requirements

- Operating system : Ubuntu
- Tools and Libraries : Python, LabelImg, Roboflow, YoloV5, OpenCV, Pytorch and Kalman Filter
- IDE Used : Google Colab
- Camera app : Droidcam

3.2 Software Specification

3.2.1 Python

Python is an interpreted, excessive-stage, widespread-purpose programming language. Created by means of guido van rossum and first launched in 1991, python's design philosophy emphasizes code readability with its extraordinary use of significant white space. Its language constructs and item-oriented approach aims to help programmers write clean, logical code for small and huge-scale projects. Python is dynamically typed and rubbish-accrued. It supports multiple programming paradigms, together with procedural, item-oriented,and purposeful programming. Python is regularly defined as a "batteries included" language due to its comprehensive preferred library.Python is meant to be an effortlessly readable language. Its formatting is visually uncluttered, and it frequently uses English key phrases in which different languages use punctuation.In contrast to many other languages, it does now not use curly brackets limit blocks, and semicolons after statements are non-compulsory. It has fewer syntactic exceptions and unique instances than c or pascal. Python uses white space indentation, as opposed to curly brackets or keywords, to delimit blocks.An growth in indentation comes after positive statements; a decrease in indentation indicates the end of the present day block. Thus, the program's visual shape appropriately represents this system's semantic shape. This feature is likewise every now and then termed the off-side rule

3.2.2 Google Colab

Google is quite aggressive in AI research. Over many years, Google developed AI framework called TensorFlow and a development tool called Colaboratory. Today TensorFlow is open-sourced and since 2017, Google made Colaboratory free for public use. Colaboratory is now known as Google Colab or simply Colab.

Another attractive feature that Google offers to the developers is the use

of GPU. Colab supports GPU and it is totally free. The reasons for making it free for public could be to make its software a standard in the academics for teaching machine learning and data science. It may also have a long term perspective of building a customer base for Google Cloud APIs which are sold per-use basis. Irrespective of the reasons, the introduction of Colab has eased the learning and development of machine learning applications.

3.2.3 LabelImg

LabelImg is a graphical image annotation tool. It is written in Python and uses Qt for its graphical interface. Annotations are saved as XML files in PASCAL VOC format, the format used by ImageNet. Besides, it also supports YOLO and CreateML formats.

Labels are used to help identify components in your data which you want to train your model to identify in datasets that are not labeled. High quality datasets are essential for computer vision and building a highly performant model. Creating computer vision models follows the garbage in, garbage out philosophy which means labeling images carefully and accurately is important.

3.2.4 Roboflow

Roboflow provides everything you need to turn images into information. We build all the tools necessary to start using computer vision, even if you're not a machine learning expert (and to supercharge your pipeline if you are). Specifically, you can use Roboflow to:

- Store and organize your image data.
- Annotate images (with the helping hand of machine learning models)
- Convert between dozens of different annotation formats.
- Understand and assess the quality of your datasets.
- Process and augment your images to help your models train faster and generalize better.

- Version control datasets and share them with your team.
- Train a model with a single click or train your own models outside of Roboflow.
- Create an infinitely scalable API to get predictions from your trained model.
- Deploy to edge devices like the NVIDIA Jetson.
- Run inference in customers' web browsers with WebGL.
- Continually improve your models over time with active learning.

3.2.5 YoloV5

YOLO is an acronym that stands for You Only Look Once. We're using Ultralytics' Version 5, which was released in June 2020 and is now the most powerful object recognition algorithm available. It's a brand-new convolutional neural network (CNN) that accurately recognises objects in real time. This method processes the entire image with a single neural network, then divides it into parts and predicts bounding boxes and probabilities for each component. The predicted probability is used to weight these bounding boxes. In the sense that it produces predictions after only one forward propagation through the neural network, the approach "looks once" at the image. After non-max suppression (which assures that the object detection algorithm is accurate), it provides discovered objects..

3.2.6 OpenCV

OpenCV is a great tool for image processing and performing computer vision tasks. It is an open-source library that can be used to perform tasks like face detection, objection tracking, landmark detection, and much more. It supports multiple languages including python, java C++. Although, For this article, we will be limiting to python only.

The library is equipped with hundreds of useful functions and algorithms, which are all freely available to us. Some of these functions are really common and are used in almost every computer vision task. Whereas many of the functions are still unexplored and haven't received much attention yet.

3.2.7 Pytorch

PyTorch is an open source machine learning library used for developing and training neural network based deep learning models. It is primarily developed by Facebook's AI research group. PyTorch can be used with Python as well as a C++. Naturally, the Python interface is more polished. Pytorch (backed by biggies like Facebook, Microsoft, SalesForce, Uber) is immensely popular in research labs. Not yet on many production servers — that are ruled by fromeworks like TensorFlow (Backed by Google) — Pytorch is picking up fast.

Unlike most other popular deep learning frameworks like TensorFlow, which use static computation graphs, PyTorch uses dynamic computation, which allows greater flexibility in building complex architectures. Pytorch uses core Python concepts like classes, structures and conditional loops — that are a lot familiar to our eyes, hence a lot more intuitive to understand. This makes it a lot simpler than other frameworks like TensorFlow that bring in their own programming style.

3.2.8 Kalman Filter

Most modern systems are equipped with numerous sensors that provide estimation of hidden (unknown) variables based on a series of measurements. For example, a GPS receiver provides location and velocity estimation, where location and velocity are the hidden variables and differential time of the satellites' signals' arrival are the measurements.

One of the biggest challenges of tracking and control systems is providing accurate and precise estimation of the hidden variables in the presence

of uncertainty. In GPS receivers, the measurement uncertainty depends on many external factors such as thermal noise, atmospheric effects, slight changes in satellite positions, receiver clock precision and many more.

The Kalman Filter is one of the most important and common estimation algorithms. The Kalman Filter produces estimates of hidden variables based on inaccurate and uncertain measurements. Also, the Kalman Filter provides a prediction of the future system state based on past estimations.

Kalman filters are used to estimate states in state space format based on linear dynamical systems. The state evolution from time k1 to time k is defined by the process model as follows:

$$x_k = Fx_{k-1} + Bu_{k-1} + w_{k-1}$$

F is the state transition matrix that has been given to the previous state vector.

$$x_{k-1}$$

B is the control-input matrix applied to the control vector

$$u_{k-1}$$

, and

$$w_{k-1}$$

is the process noise vector that is assumed to be zero-mean Gaussian with the covariance Q.

The process model is paired with the measurement model that describes the relationship between the state and the measurement at the current time step k as:

$$z_k = Hx_k + v_k$$

where

$$z_k$$

is the measurement vector, H is the measurement matrix, and

$$v_k$$

is the measurement noise vector that is assumed to be zero-mean Gaussian with the covariance R, Note that sometimes the term "measurement" is called "observation" in different literature.

The role of the Kalman filter is to provide estimate of

$$x_k$$

at time k., given the initial estimate of

$$x_0$$

, the series of measurement,

$$z_1, z_2, \ldots, z_k$$

, and the information of the system described by F, B, H, Q, and R. Subscripts to these matrices are removed here since they are assumed to be invariant across time, as they are in most applications. Although the covariance matrices are designed to reflect the statistics of the noises, in many practical situations, the true statistics of the noises are unknown or not Gaussian. As a result, Q and R are frequently employed as tuning factors that the user can tweak to achieve the desired result.

Transition and observation matrix in the Kalman filter for Python is implemented in the following way.

```
1 from pykalman import KalmanFilter
2 >>> import numpy as np
3 >>> kf = KalmanFilter(transition_matrices = [[1, 1], [0,
  1]], observation_matrices = [[0.1, 0.5], [-0.3, 0.0]])
4 >>> measurements = np.asarray([[1,0], [0,0], [0,1]])  # 3
  observations
5 >>> kf = kf.em(measurements, n_iter=5)
6 >>> (filtered_state_means, filtered_state_covariances) =
  kf.filter(measurements)
```

Fig. 3.1: Sample code - Kalman Filter Implementation

For example, if

$$A_{k-1}$$

is an nxn matrix then the argument transition_matrices is given as transition_matrices = [[row 1],[row 2],....[row n]].

The argument observation_matrices is also specified in the similar manner.

Applications of Kalman Filter

- Vehicle navigation and guidance, particularly for aviation and spacecraft.
- Robotic trajectory planning and correction.
- Position awareness radar sensors for driverless vehicles' advanced driver assistance systems (ADAS).
- Stabilizing depth measurements, object tracking (e.g., faces, heads, hands), fusing data from laser scanners, stereo cameras for depth and velocity measurements, and 3D mapping with Kinect or range cameras are just a few of the computer vision applications.

3.3 Problem Definition

We could look into a variety of approaches and models to handle the challenge of quick tennis ball recognition tracking. However, if we want a model

that can be used to solve real-world problems, we need presumably define the following model requirements:

- The model should perform well in the domain of interest, such as on real-life sports videos.
- The model should be able to work in real time in many cases, since you may want to handle a video stream without any noticeable delays.

With these constraints in mind, we can narrow the range of viable models. Simple OpenCV blob identification algorithms, for example, will not attain the required performance due to the tough environment. Deep learning models like Faster R-CNN and Detectron2, on the other hand, cannot work in real time, even if they can attain the requisite performance.

Overall, we feel that the YOLO family of requirements models offer the optimal trade-off between computational burden and accuracy.

3.4 Purpose

The ball is an important component of numerous sports, including soccer, golf, tennis, and football, and it receives a lot of attention. As a result, there is no doubt that developing an automated ball tracking system is an important step in developing a strong Sports AI.

The challenge of automated ball tracking can be expressed as follows: at each image frame from a video, we must detect whether there is a ball in the image and, if so, where it is located. Object detection is the term used in the computer vision community to describe this challenge.

Due to the speed of the ball and the fact that it is frequently obscured on recordings, one of the top difficulties to address with artificial intelligence in sports is ball detection.

3.5 Functional Requirements

The system should be able to detect the tennis ball from live video from camera. And from the input video the system should be able to track and predict the trajectory of tennis ball.

3.6 Performance Requirements

The system would need least 8 GB of RAM and Nvidia RTX 3050 GPU. Less RAM and GPU will result in the poor performance of the system. More number of CUDA cores means more data can be processed parallelly. More clock speed means that a single core can perform much faster. The GPUs get better with new generations and architectures, so a graphic card with more number of CUDA cores is not necessarily more powerful than the one with lesser CUDA cores.

3.7 Constraints

There are certain constraints regarding the usage of these system as we assume that user must have a good quality mobile camera or digital camera with the quality of at least able to record 30 FPS or above videos for the analysis. Thus the application is subject to have a good quality mobile camera or digital camera. The low-quality videos with low-quality cameras will lead to issues with accurate predictions of the videos. Also system requires more light in the surrounding.

CHAPTER 4
SYSTEM ANALYSIS

By definition, system analysis is a method of gathering information, understanding the facts, diagnosing the problem, and using the information to either create a completely new framework or prescribe improvements to an existing framework.

A good system inquiry is examining a business situation with the goal of improving it through improved strategies and procedures. The inquiry stage defines the framework's requirements and the challenges that the client is attempting to explain, regardless of how the requirements will be implemented.

4.1 Requirements Analysis

Requirements analysis, often known as requirements engineering, is the process of determining what customers want from a new or modified product. These highlights, also known as requirements, must be quantitative, meaningful, and listed. Such requirements are commonly referred to be helpful judgments in programming design.

4.2 Existing System

In order to provide an algorithm that exceeds prior algorithms in terms of accuracy and speed, a reliable and efficient ball tracking system combines the advantages of computer vision and machine learning approaches. The fundamental idea is to create a video stabilisation framework before using random forest classification to separate ball candidates in the movie. Com-

bine the saliency map of the frame with the colour of the ball for successful ball candidate segmentation in video frames.

Limitations of Existing Systems:

- For ball detection, some systems employed OpenCV. The colour of the ball is used to detect it. When objects of the same colour appear in the backdrop, the algorithm recognises them as balls.
- When it comes to real-time detection, RCNN and SSD-based systems do not perform better and cannot provide FPS of 30 or higher.

4.3 Proposed System

The main goal of the proposed system is to detect, track, and predict trajectory in real-time 2D video. The footage is sent to the system via a mobile camera and the droidcam app, which enables for high-speed capturing. The device can track tennis balls with a frame rate of 30 FPS or higher. Because the YoloV5 model was used to train the model. YoloV5 outperforms Yolo models and other models like SSD and RCNN when it comes to real-time detection. Also, by using a 3D camera capable of collecting quick movements and providing a higher FPS rate, this system can be turned to a 3D system. It can be used for tennis game analysis, player analysis, and so on using the current 2D system by upgrading the system.

Advantages of Proposed Systems:

- The suggested technology detects tennis balls in videos with a frame rate of 30 FPS or higher. This permits the tennis ball to be detected in motion.
- The prediction is made using the kalman filter and a list containing the centroid of the bounding box.

4.4 Feasibility Study

A feasibility study is a high-level container for the entire System Analysis and Design Process. The examination begins with the problem definition being grouped. The goal of feasibility is to determine whether or not the work is worth attempting. The expert creates a valid model of the framework after creating an acknowledgment issue definition. A quest for options is meticulously dissected. The feasibility study is divided into three sections.

4.4.1 Technical Feasibility

Technical feasibility is a study of function performance and requirements that may affect the ability to build a suitable framework. It is frequently the most difficult place to assess at this stage of the framework advancement procedure. During specialised examination, the investigator evaluates the specialised benefits of the system concept while collecting additional data on performance, reliability, viability, and reducibility. The most common specialised issues raised during the attainability phase of examination include. Before beginning the task, understand the many progressions associated with the suggested framework. The framework must be clear about what advancements are necessary for the improvement of the new framework. The new system can be implemented with current technical resources. This project is technically viable when these facts are considered.

4.4.2 Operational Feasibility

Only if the project can be translated into information systems that meet the needs will it be lucrative. In a nutshell, this crash test determines whether the system will function after it has been built and installed. Here are some criteria for evaluating the project's success:

- There may be opposition if the old system is so popular and widely used that people are unaware of the grounds for change.
- If the user cannot accept the existing system approaches. Users can then adopt improvements that will make systems more efficient.
- Early involvement reduces the risk of programme failure and, in general, increases the likelihood of a successful system. Because the method presented was effective in reducing the problem,

4.4.3 Economical Feasibility

Cost-benefit analysis is one of the most important pieces of information in a feasibility study, which is an evaluation of economic justification or a PC-based framework project cost-benefit-analysis investigation depicts costs for project advancement and compares them to tangible and intangible system costs. To establish a system that is genuinely used, it requires monetary benefits or, on the other hand, expenses that exceed or are made equivalent. The questions posed with the intention of evaluating are

- The cost of conducting a comprehensive system inquiry
- Hardware and software costs for the application class in question
- Benefits such as lower expenses or fewer costly errors

CHAPTER 5
SYSTEM DESIGN

5.1 Applications Architecture

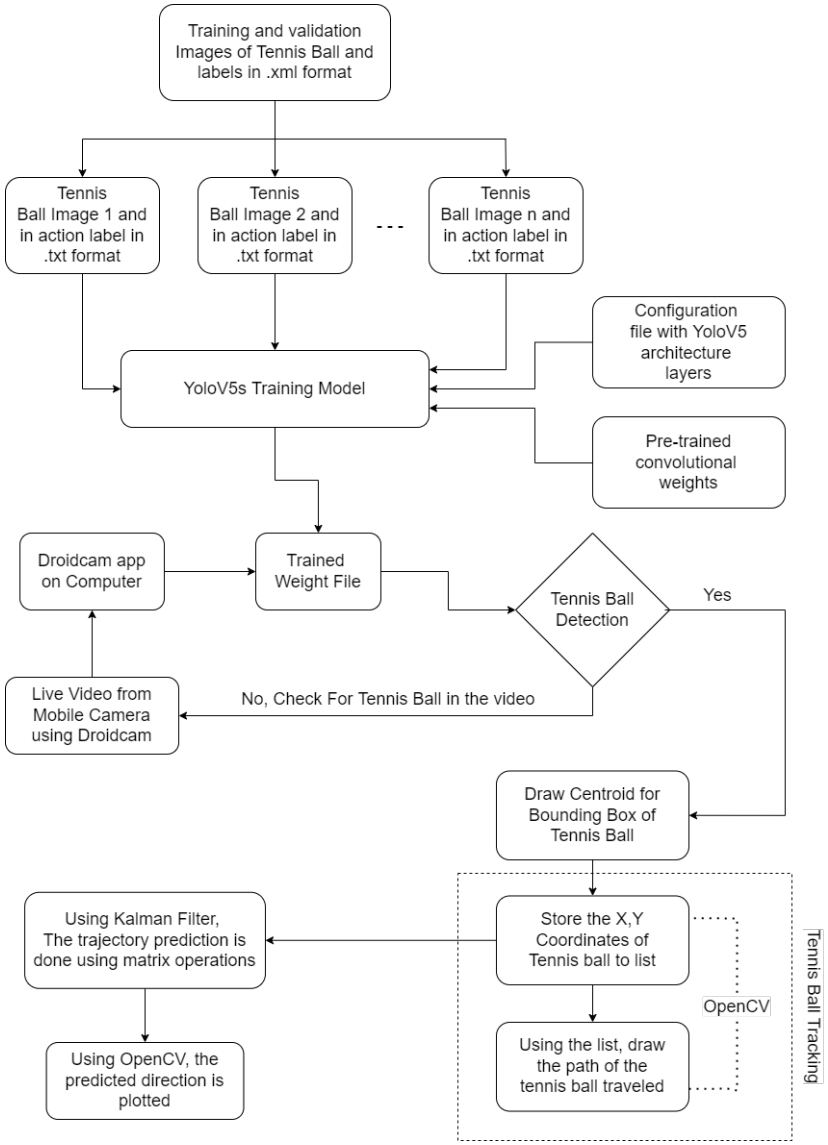


Fig. 5.1: Architecture Diagram

Fig. 4.1 is the architecture of the proposed system, A Real-Time System for Tennis Ball Tracking and Trajectory Prediction. In the developed system, the live video is given to the computer from mobile camera using Droidcam application. The live video is then used with trained YoloV5s weights to detect tennis ball. And the coordinates are stored in list. Using OpenCV the centroid and the path of the ball is drawn. Then, stored list value is applied with kalman filter to predict the direction of the tennis ball in the future.

5.2 List of Modules

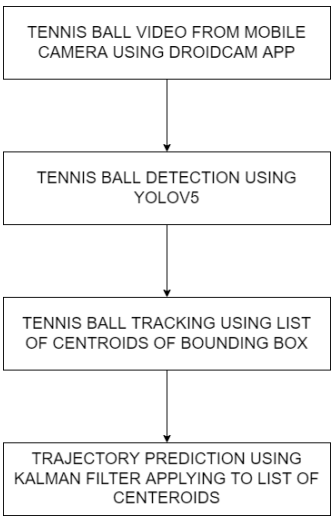


Fig. 5.2: Process Flow Diagram

A Real-Time System for Tennis Ball Tracking and Trajectory Prediction consists of mainly three modules. They are:

- Tennis Ball Detection
- Tennis Ball Tracking
- Tennis Ball Trajectory Prediction

5.2.1 Tennis Ball Detection

Model Trained using Yolov5s model and saved the best weights. The weights that have been saved are then used to detect the tennis ball from the camera.

For a greater frame rate than a standard laptop webcam, a mobile camera was used. The droidcam app is used to attach the mobile camera to the computer.

5.2.2 Tennis Ball Tracking

At each frame the centroid for tennis ball's bounding box is calculated. Using "cv2.circle" plotted the centroid of the Tennis ball and saved the coordinates to a list. With the help of saved coordinates in the list, the trajectory of tennis ball is created using "cv2.line".

5.2.3 Tennis Ball Trajectory Prediction

Using Kalman filter, the trajectory prediction is done. Kalman filter is an algorithm that takes measurements over time and creates a prediction of the next measurements. With the help of the saved coordinates list for tracking the tennis ball, the prediction points are measured for the x,y coordinates.

CHAPTER 6
SYSTEM IMPLEMENTATION

6.1 Dataset Collection

1000+ tennis ball images are captured using mobile camera. The images include fast movement images of tennis ball. The images are captured in a manner, different light conditions Labeled using LabelImg.Then image augmentation (mosaic) applied to tennis ball images using Roboflow. Also, tennis ball images in dataset increased to 1220 images (1100 images in Training Set and 120 images in Validation Set.)

Roboflow is a developer framework for better data collecting, preprocessing, and model training procedures in computer vision. Roboflow provides customers with access to public datasets as well as the ability to submit their own bespoke data. Roboflow supports a number of annotation formats. Image orientations, resizing, contrasting, and data augmentations are all processes in the data pre-processing process.

Within the framework, the complete workflow can be coordinated by teams. EfficientNet, MobileNet, Yolo, TensorFlow, PyTorch, and more model libraries are already available for model training. Following that, model deployment and visualisation choices are offered, effectively covering the whole state-of-the-art.

Fig. 6.1: Tennis Ball in Different Distances and Background Light

6.2 LabelImg and Roboflow

Figure 6.2 shows annotating tennis ball using labelimg. Using labelimg created bounding for tennis ball for training the model using YoloV5 model. When annotating image using labelimg, the coordinates of bounding box, number and name of class are stored in a .txt file. The required details for training Yolo model is stored on to a .xml file.

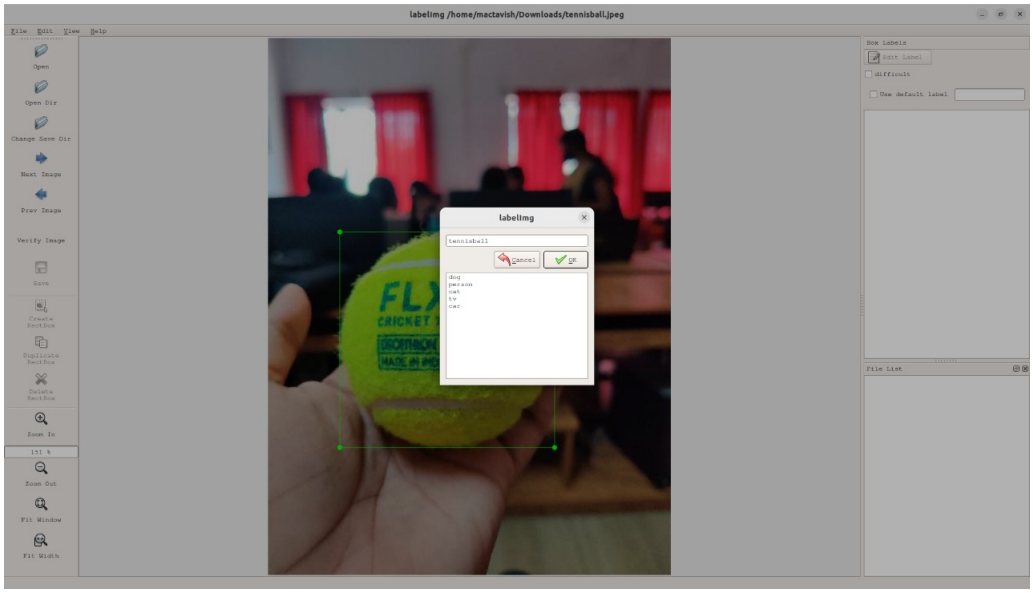


Fig. 6.2: Tennis Ball image Annotation using LabelImg

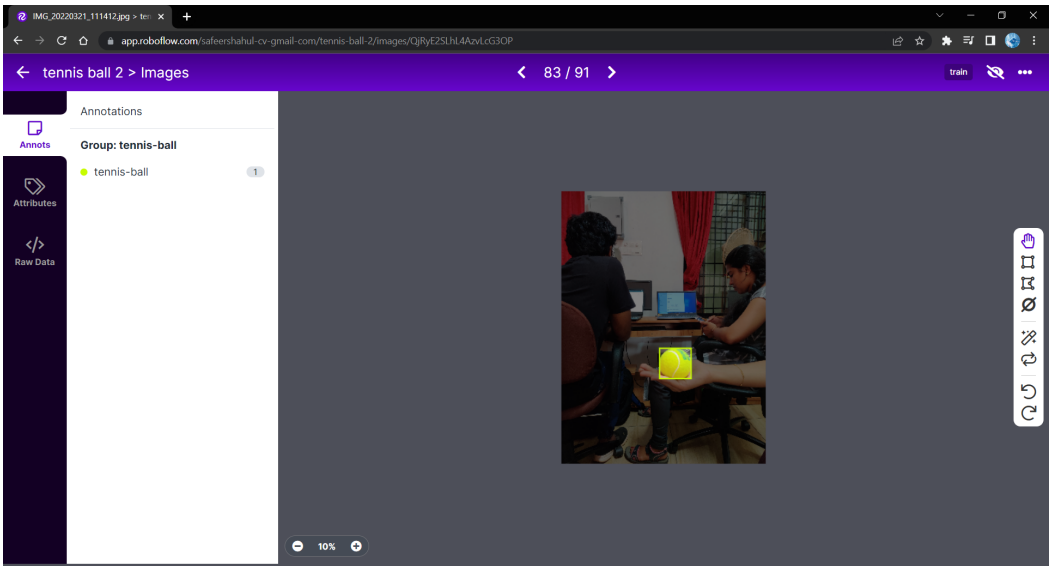


Fig. 6.3: Tennis Ball dataset loaded to Roboflow.

Using Roboflow, we can load our annotated tennis ball dataset into it and we can load the class and label files to it. If we have not annotated the image, we can use Roboflow to annotate image.

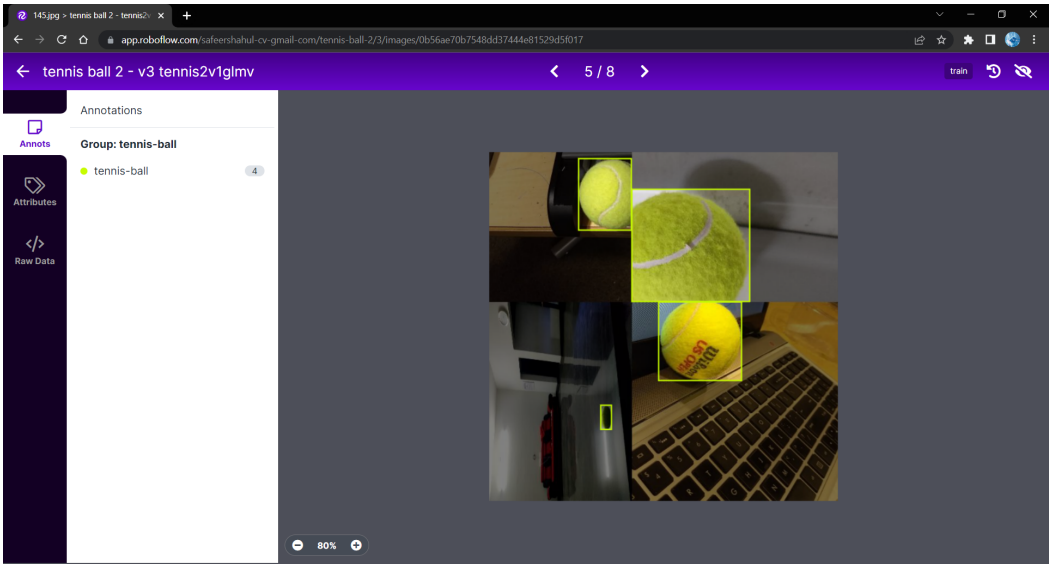


Fig. 6.4: Tennis Ball dataset augmented to mosaic using Roboflow.

Figure 6.4 shows the tennis ball images in dataset is augmented to mosaic. The Roboflow allows users to augment to different types. Like, saturation,crop, flip, rotate and so on. the augmentation is used accordance with features of the object. If the object is tiny, augmentation like mosaic and flip-rotate are used.

6.3 DroidCam

The DroidCam app on mobile phone and DroidCam client app on computer used for connecting mobile camera to computer. DroidCam client app allows to zoom in/out, auto focus and other camera controls within computer using droidcam client app.

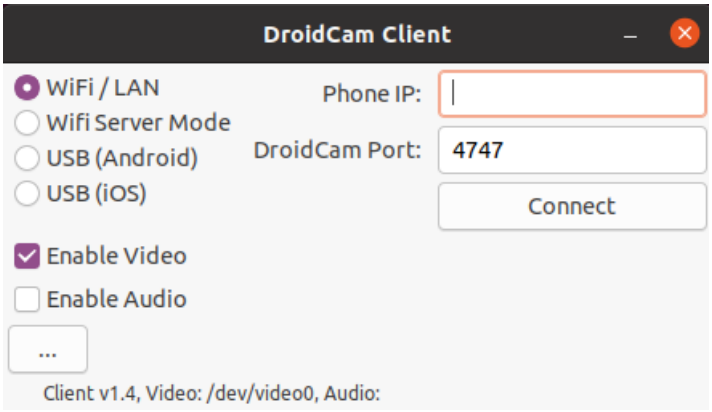


Fig. 6.5: DroidCam Interface - Ubuntu.

6.4 Google Colab

Fig 6.6 shows the screenshot of my Google colab account. Google colab used for training custom tennis ball detection model using YoloV5. Google colab provide maximum capacity GPU for producing output(trained weights) at maximum speed.

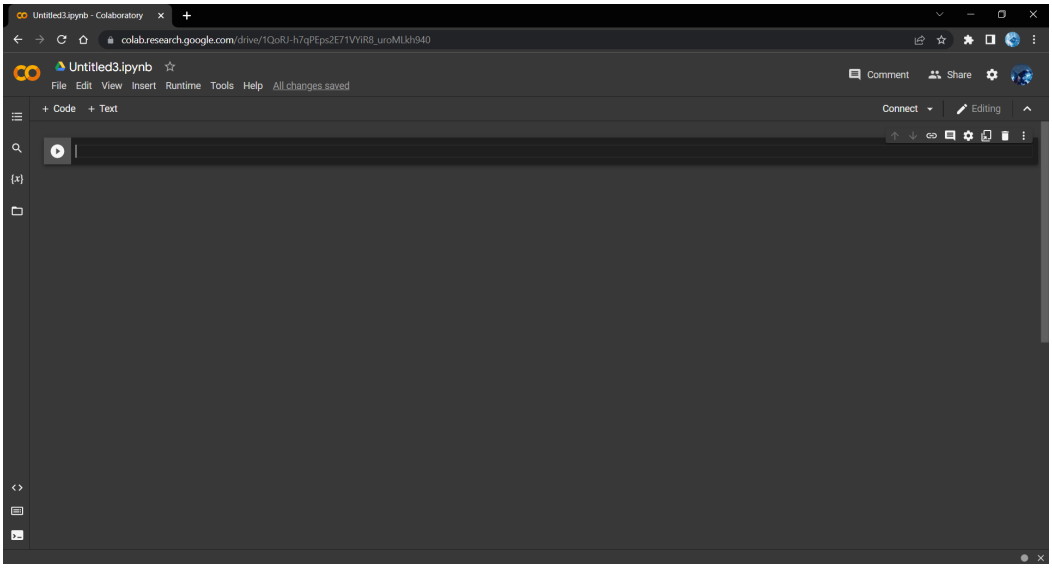


Fig. 6.6: Google Colab Account.

6.5 TensorBoard

TensorBoard is a tool that provides the measurements and visualisations required for machine learning. It allows you to track experiment parameters such as loss and accuracy, visualise the model graph, project embeddings into a lower-dimensional space, and much more.

Using TensorBoard, analyzed the tennis ball detection model's precision, recall, mAP and loss. The figure 6.7 show TensorBoard interface:

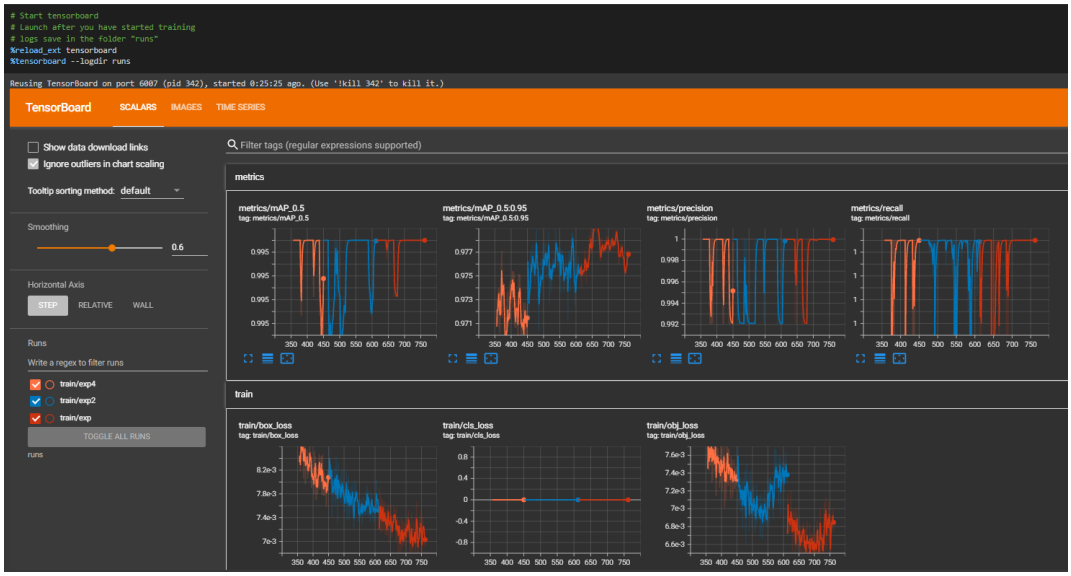


Fig. 6.7: TensorBoard Interface.

CHAPTER 7
RESULTS AND DISCUSSION

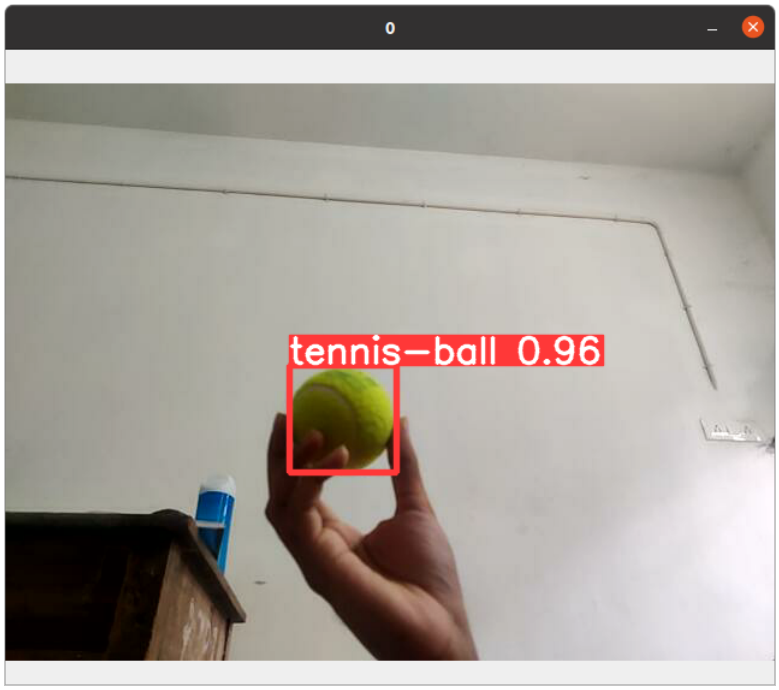


Fig. 7.1: Tennis Ball Detection in Real-Time.

The mobile camera is connected to the computer via Droidcam. Figure 7.1 shows the detection using the learned custom tennis ball weights.

Code for running detect.py:

python3 detect.py –weights tennis_custom_weights.pt –source 0
here, weights - custom trained weights of tennis ball using YoloV5s model.
source - webcam(default 0) or source video path(tennis ball video).

Fig. 7.2: Tennis Ball Tracking Using OpenCV

Then locate the Tennis Ball's X,Y coordinates, as well as its centroid, and save them to a list. Using OpenCV, draw the real path of the tennis ball and the centroid of the tennis ball's bounding box using the list of centroids.

Fig. 7.3: Prediction of Tennis Ball Trajectory using Kalman Filter

The direction of tennis is predicted with measurable accuracy using the Kalman filter (i.e, stored centroid list values). The kalman filter uses the uncertainties of observed values and applies matrix operation on measured

and unknown data to predict the tennis ball direction when applied to measured estimations of time.

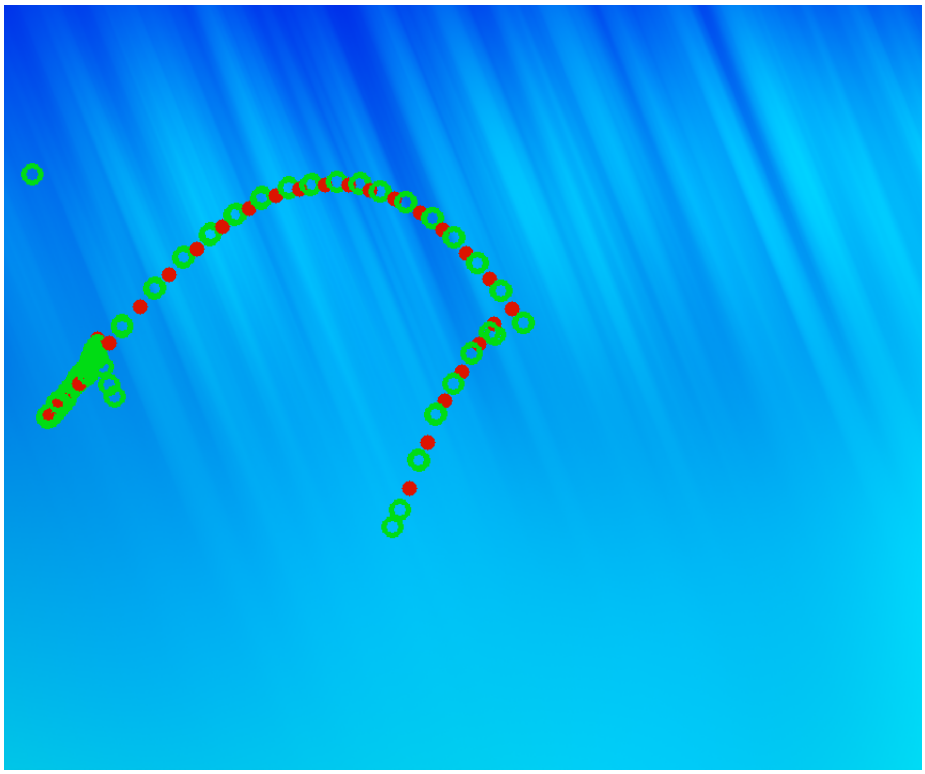


Fig. 7.4: Plot - Tennis Ball actual and predicted path

The stored centroid list values are saved to a .json file. Same way, the predicted values are stored in to a .json file. using the two .json files, the actual and predicted values are plotted to a image.

7.1 Evaluation Measures

Measurement is essential for us to understand the external world and through millions of years of life, we have developed a sense of measurement. It helps to understand the efficiency of our project. Some of the evaluation measures that I use to check the efficiency of my project is Mean Average Precision (mAP), Precision and Recall. Used Tensorboard for finding evaluation metrics.

7.1.1 Mean Average Precision (mAP)

To evaluate object detection models like R-CNN and YOLO, the mean average precision (mAP) is used. The mAP compares the ground-truth bounding box to the detected box and returns a score. The higher the score, the more accurate the model is in its detections.

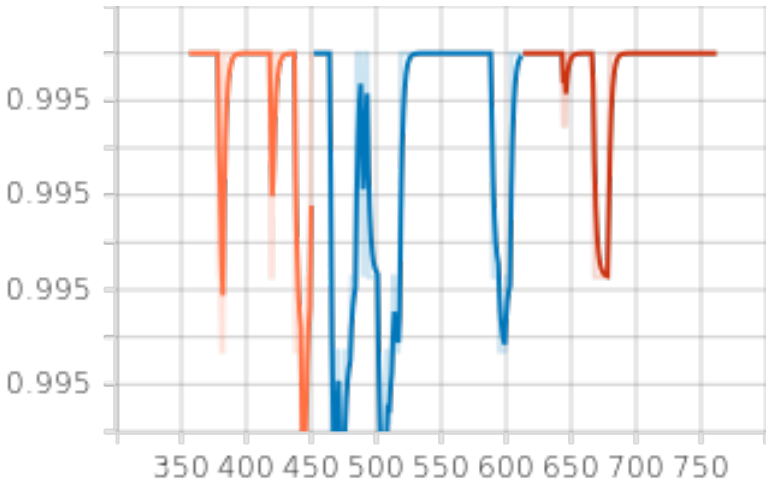


Fig. 7.5: metrics - mAP - 0.5

We evaluate the mAP averaged for IoU in [0.5 : 0.05 : 0.95] (COCO's standard metric, simply denoted as mAP@[.5, .95]) and mAP@0.5 (PASCAL VOC's metric).

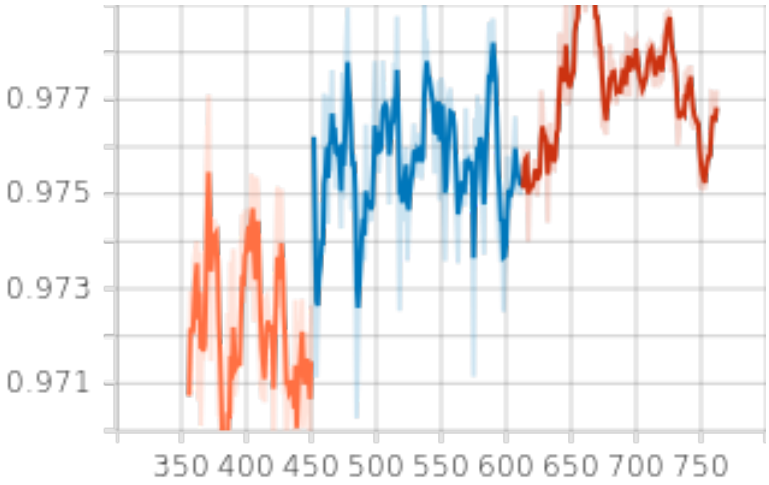


Fig. 7.6: metrics - mAP - 0.5:0.95

7.1.2 Precision

The ability of a classification model to identify only the relevant data points. Mathematically, precision the number of true positives divided by the number of true positives plus the number of false positives.

$$Precision = \frac{TruePositive}{TruePositive + FalsePositive}$$

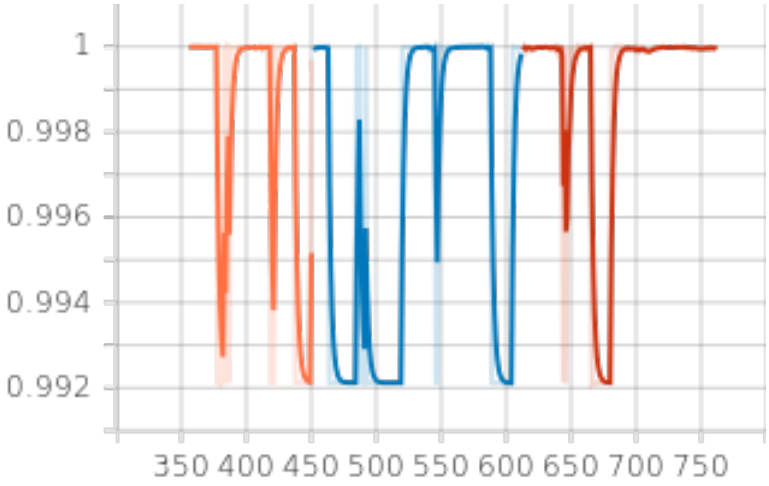


Fig. 7.7: Precision

When compared to the training labels, a system with high recall but low precision returns a large number of results, but the majority of its projected labels are inaccurate. When compared to the training labels, a system with high precision but low recall returns very few results, but the majority of its projected labels are correct. A perfect system with great precision and recall will return a large number of results, all of which will be correctly labelled.

7.1.3 Recall

The ability of a model to find all the relevant cases within a data set. Mathematically, we define recall as the number of true positives divided by the number of true positives plus the number of false negatives.

$$Recall = \frac{TruePositive}{TruePositive + FalseNegative}$$

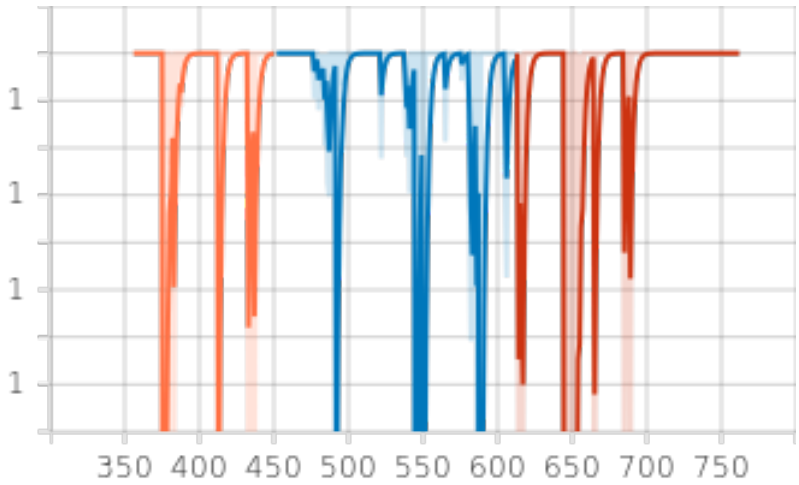


Fig. 7.8: Recall

CHAPTER 8
CONCLUSION

This project is mostly focused on real-time 2D videos of tennis ball. Tennis ball detection, tracking, and trajectory prediction are performed using 2D real-time videos. The YoloV5 model has a maximum FPS of 150. So, the present build system can be utilised on a high-performance computer with a GPU with the most cuda cores (i.e., a system with the most cuda cores improves graphical processing and makes multi-threading easier) and high-frame-rate cameras for additional analysis during the tennis game. For example, checking for ball in/out, player fitness analysis, and so on. The present method is designed mostly for recreational tennis games.

Previously, such systems used machine learning techniques like RCNN and SSD to detect and track tennis balls. However, unlike the Yolo model, SSD and RCNN do not allow for real-time detection. SSD outperforms Yolo in terms of precision. However, SSD is not as quick as Yolo. Also, computers are unable to recognise fast moving objects in real time with a higher FPS. The suggested approach provides for faster tennis ball recognition as well as easy tracking and prediction of the tennis ball's trajectory. This technique can also be used with a 3D camera for a simpler and faster output. We attempted this with the ZED 2 3D camera. However, the ZED camera does not identify fast moving objects, and the FPS is extremely low. We attempted this with the ZED 2 3D camera. However, the ZED camera does not identify fast moving objects and has a poor FPS.

CHAPTER 9
SCOPE FOR THE FUTURE ENHANCEMENT

The proposed method is extremely adaptable to future improvements. On 3D real-time videos, this system can be expanded to do detection, tracking, and trajectory prediction. The 3D method can also be utilised to construct a Tennis Playing Robot, allowing for faster and better game analysis. We can acquire a lot of game data from a 3D perspective. Within a fraction of time, for example, the speed of the smash, the ball, the player, the distance travelled by the player, the ball, and so on.

This is only possible if we have a system with the highest graphical processing power, as well as a 3D camera with the highest FPS and Megapixels. If we need to employ two 3D cameras, the system should also be able to multi-thread. Otherwise, the process will not be loaded by the system.

If we construct a 3D video system, it can be used in both casual and professional games. This technique can also be used to coach tennis players and to create a system to read player statistics after a game.

BIBLIOGRAPHY

[1] Qazi, T., Mukherjee, P., Srivastava, S., Lall, B., & Chauhan, N. R.*"Automated ball tracking in tennis videos,"* 2015 Third International Conference on Image Information Processing (ICIIP) (pp. 236-240). IEEE.

[2] Chakraborty, B., & Meher, S. (2013, December)*"A trajectory-based ball detection and tracking system with applications to shooting angle and velocity estimation in basketball videos,"* 2013 Annual IEEE India Conference (INDICON) (pp. 1-6). IEEE

[3] Vinyes Mora, S. (2018).*"Computer vision and machine learning for in-play tennis analysis: framework, algorithms and implementation,"* (Doctoral dissertation, Imperial College London).

[4] X. Yu, H. W. Leong, C. Xu, & Q. Tian*"Trajectory-Based Ball Detection and Tracking in Broadcast Soccer Video,"* IEEE Transactions on Multimedia, vol. 8, no. 6, pp. 1164–1178, 2006.

[5] Ferede, S., Xie, X., Zhang, C., Du, J., & Shi, G. (2020, October),*"Small Ball Tracking with Trajectory Prediction"*, In 2020 IEEE 5th International Conference on Signal and Image Processing (ICSIP) (pp. 215-219). IEEE.

[6] Sun, N. E., Lin, Y. C., Chuang, S. P., Hsu, T. H., Yu, D. R., Chung, H. Y., & İk, T. U. (2020, December),*"TrackNetV2: Efficient Shuttlecock Tracking Network"*, In 2020 International Conference on Pervasive Artificial Intelligence (ICPAI) (pp. 86-91). IEEE.

[7] https://github.com/ultralytics/yolov5

[8] https://pykalman.github.io/